(19) Europäisches Pat ntamt
Europ an Patent Offic
Offic europé n d s brev ts

(11) Publication number: 0 157 303 A2

(12) # EUROPEAN PATENT APPLICATION

(21) Application number: 85103322.5

(22) Date of filing: 21.03.85

(51) Int. Cl.⁴: G 06 F 1/00, G 06 F 12/14

(30) Priority: 31.03.84 JP 63618/84

(43) Date of publication of application: 09.10.85 Bulletin 85/41

(84) Designated Contracting States: DE FR GB

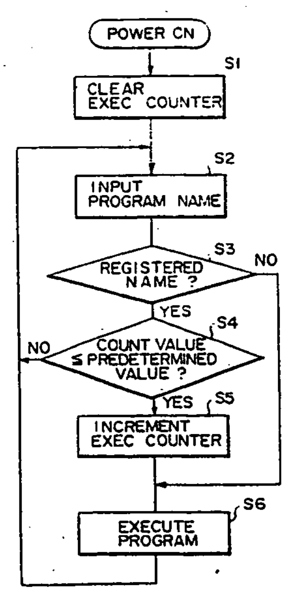
(71) Applicant: Kabushiki Kaisha Toshiba, 72, Horikawa-cho Saiwai-ku, Kawasaki-shi Kanagawa-ken 210 (JP)

(72) Inventor: Tamada, Masuo c/o Patent Division, K.K. Toshiba 1-1 Shibaura 1-chome, Minato-ku Tokyo 105 (JP)
Inventor: Kokuryo, Hitoshi c/o Patent Division, K.K. Toshiba 1-1 Shibaura 1-chome, Minato-ku Tokyo 105 (JP)
Inventor: Tamura, Shinsuke c/o Patent Divison, K.K. Toshiba 1-1 Shibaura 1-chome, Minato-ku Tokyo 105 (JP)
Inventor: Ozaki, Hiroshi c/o Patent Divison, K.K. Toshiba 1-1 Shibaura 1-chome, Minato-ku Tokyo 105 (JP)

(74) Representative: Henkel, Feller, Hänzel & Partner, Möhlstrasse 37, D-8000 München 80 (DE)

(54) **Data processing device.**

(57) An IC card comprises CPU (33), PROM (34) for storing programs, and RAM (35) for storing the results of arithmetic operations. PROM (34) stores a table on which the names of sp cific programs are registered. RAM (35) has execution count rs which counts up each time each program is run so that the number of times the specific program is run is memorized. Prior to the running of each program by CPU (33), it is judged whether or not the program is a specific one. If it is the specific program, the execution counter counts up. When the count value of the execution counter is above a predetermined value, the running of the specific program is prevented.



EP 0 157 303 A2

Data processing device

This invention relates to a data processing device, and more particularly, to the protection of a specific program run in a portable data processing device such as an IC card which includes a microcomputer, ROM, RAM, etc.

Banks have recently issued IC cards which have a RAM for storing an account number and transaction data, a ROM for storing a control program, and a microprocessor. The ROM memorizes a password of the card. Only when the password input in a keyboard of a bank terminal unit coincides with that stored in the data processing device, can the card be used, so that illegal use of the card by a third person is prevented. However, illegal use by its owner is not prevented. For example, the ROM stores a coding process program for the communucation between the IC card and the host computer, which must be kept secret from the user (owner of the card). The IC card stores many other secret programs as well. The algorithm of these programs can be discovered if they are run millions of times or more. It has so far been impossible to completely prevent illegal use of the IC card by the owner through the discovery of the algorithms of the programs.

The object of the invention is to provide a data processing device wherein the algorithm of a sp cific program cannot be discovered. This object can be

achieved by limiting the number of times the specific program is run, prolonging the time taken in running the specific program, preventing the continuous running of the specific program, and/or preventing each program from being run in an order which is not predetermined.

This invention can be more fully understood from the following detailed description when taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a perspective view showing a terminal apparatus used in a bank for use with an IC card according to a first embodiment of the data processing apparatus of the present invention;

Fig. 2 is a block diagram of the terminal apparatus shown in Fig. 1;

Fig. 3 is a block diagram of the IC card according to the first embodiment;

Fig. 4 is a flow chart of the operation of the first embodiment;

Fig. 5 is a flow chart illustrating the operation of a second embodiment of this invention;

Fig. 6 is a flow chart illustrating the operation of a third embodiment of this invention;

Fig. 7A shows an example of an execution order table Ep used in a fourth embodiment of this invention;

Fig. 7B shows an example of register Ef for registering the name of the program which was formerly run and register En for registering the name of the program which is presently being run, as used in the fourth embodiment; and

Fig. 8 is a flow chart showing the operation of the fourth embodiment.

A first embodiment of the data processing device according to the present invention will be described with reference to the accompanying drawings.

Banking facilities have recently issued IC cards in lieu of bankbooks, which memorize the account number and transaction data. The IC card is used in

this description as an example of the data processing device. The IC card operates with a bank terminal unit shown in Figs. 1 and 2 to perform transactions.

Fig. 1 perspectively shows the bank terminal unit. Fig. 2 shows a block circuit thereof. Terminal unit 11 has CPU 21 for its entire control. CPU 21 is connected to ROM 22 storing control programs and to RAM 23 for storing data. Also, CPU 21 is connected to keyboard 24, CRT display 25, printer 26, floppy disc unit 27, IC card reader/writer 28 through interface 29, and modem 31 through interface 32. IC card reader/writer 28 reads and writes data in IC card 12 inserted through card insertion opening 28'. Modem 31 modulates the data fed from CPU 21. By using telephone 30, the modulated data are fed through telephone line 13 to a host computer, etc. Modem 31 demodulates the data fed from the host computer, etc. through line 13 and supplies them to CPU 21. The data communication between CPU 21 and the host computer is performed by coding the data to keep the data secret. A program for producing coding key data used in the coding process is stored in IC card 12, and will be described later.

Fig. 3 shows the structure of IC card 12. CPU 33, which is a microcomputer, is connected to PROM 34 which memorizes control programs, e.g., for producing the above-mentioned secret coding key data producing program necessary for the coding process (hereinafter, a program which must be kept secret from the user is referred to as a "specific program") and for limiting the running of this specific program. PROM 34 also memorizes a password assigned to the owner of the IC card. CPU 33 is also connected to RAM 35 for storing data and to connector 36 which is coupled to a connecting pin (not shown) provided within IC card reader/writer 28.

A system comprising the IC card and the bank terminal unit is generally operated in the following manner. IC card 12 is inserted into insertion

opening 28'. Connector 36 of IC card 12 is mechanically and electrically connected to the connecting pin in IC card reader/writer 28. The password is input from keyboard 24, and is fed to CPU 33 of IC card 12 through CPU 21, interface 29 and IC card reader/writer 28. The password is then compared with the password stored in PROM 34.

The operation for limiting the number of times specific programs are run, which is essential to the invention, will now be described. Fig. 4 illustrates the operation of the first embodiment of the invention. In addition to the above-described structure, in the first embodiment, PROM 34 has tables registering the names of specific programs, and RAM 35 has execution counters which count and memorize the number of times the corresponding specific programs are run.

When the power is turned on, the execution counter is cleared in Step 1. In Step 2, the name of a program to be run is input. In Step 3, the name of the input program is compared with that of the programs registered in the tables to judge whether or not the program to be run is the specific program. If it is not the specific program, it is run in Step 6. Then, the operation returns from Step 6 to Step 2 to wait for the input of the name of the next program.

When the program to be run is judged to be the specific one, Step 4 judges whether or not the count value of the execution counter for that program is below a predetermined value. If the value of the execution counter is below or equal to the predetermined value, "1" is counted in the execution counter in Step 5, and the routine goes to Step 6 where the program is executed. If the value of the execution counter is above the predetermined value, the program is not run, and the operation returns from Step 4 to Step 2 to wait for the input of the name of the next program. The predetermined value, which is compared with the count

value of the execution counter and indicates the maximun number of times the specific program is allowed to be run, is set above the value at which the specific program is normally run from the turn-on to turn-off of the power and far below the value at which the algorithm of the specific program may be discovered.

In the first embodiment, unless the power of the terminal unit is turned off and then on, the number of times the specific program is run is limited. Thus, it is impossible to execute the specific program millions of times or more to discover the algorithm of the specific program. However, as stated above, the execution counter is cleared when the power is turned on. If the power of the terminal unit is turned off and then turned on again to clear the execution counter, the predetermined number of times may be increased indefinitely. This problem can be eliminated if the count value in the execution counter is stored in PROM 34 before the power is turned off and the execution counter is initialized based on the data in PROM 34 each time the power is turned on again.

The description of Fig. 4 does not refer to the comparison of the password. Generally, the comparison of the password is performed right before the running of the program in Step 6.

A second embodiment of the invention will be described hereinafter. Fig. 5 shows the operation of the second embodiment. The IC card of the second embodiment has PROM 35 for storing tables registering the names of specific programs and tables memorizing a predetermined delay time for each specific program.

When the power is turned on, the name of a program to be run is input in Step 11. In Step 12, it is judged whether the input program is the specific program. If it is not the specific program, the program is run in Step 14. Then, the operation returns to Step 11 to wait for the input of the name of the next program. On the

other hand, if the input program is the specific one, a delay time operation is executed in Step 13 before the program is run in Step 14. The delay time is designed to be short in consideration of the normal operation. However, if the delay time operation is repeated millions of times or more, the total delay time becomes immense so that it is, in fact, impossible to run the specific program many times to discover the algorithm of the specific program.

In Fig. 5, the delay time operation is executed before the specific program is run. However, it is possible to execute the delay time operation while or after the specific program is run. In addition, it is possible to register the execution position of the delay time program in the above-mentioned registration table to change the execution position of the delay time program according to each specific program.

Fig. 6 shows a third embodiment of the invention, which is designed to prohibit the continuous running of the specific program. As in the first embodiment, PROM 34 has tables registering the names of specific programs, and RAM 35 has execution counters which count and memorize the number of times the corresponding specific program is run.

When the power is turned on, the execution counters are cleared in Step 21. In Step 22, the name of a program to be run is input. In Step 23, it is judged whether or not the input program is a specific one. If the input program is not the specific one, it is run in Step 27. Then, the operation returns from Step 27 to Step 22 to wait for the input of the name of the next program. If the input program is the specific one, Step 24 judges whether or not the count value of the execution counter is "1". If the count value is not "1," "1" is counted in the counter in Step 25, and the execution counters excluding that of the present specific program are cleared. Thereafter, the routine

goes to Step 27 where the program is executed. On the other hand, if the count value is "1," the operation returns to Step 22 to wait for the input of the name of the next program.

In the third embodiment, value "1" is set in the execution counter right before the specific program is run, and the execution counter of the specific program is cleared right before another program is run. Therefore, the same specific program cannot be continuously run. The repeated running of the same specific program inevitably involves the running of other intervening specific programs. Twice the normal time is required to repeat the running of the specific prgram. Thus, it is impossible to discover the algorithm of the specific program.

Next, a fourth embodiment will be described. In the fourth embodiment, the order of executing each program is preset. Because the execution of the programs in a different order is impossible, the continuous running of the same program is prevented. To this end, execution order table Ep as shown in Fig. 7A is stored in PROM 34 of IC card 12. Also, RAM 35 has register Ef for registering the name of the formerly run program (or formerly run program name register Ef) and register En for registering the name of the presently run program (or presently run program name register En) as shown in Fig. 7B. Execution order table Ep registers program names O, A, B, C and D in an order which allows the programs to be run. Because the programs can be run only in the order registered in the execution order table Ep, the repeated running of a program necessarily involves the running of another program and takes a great amount of time. Thus, it is, in fact, impossible to discover algorithm by repeating the program.

The above operation will be described in greater detail with reference to Fig. 8. When the power is

turned on, formerly run program name register Ef is cleared in Step 31. In Step 32, the name of the program to be run is input to presently run program name register En. Step 33 judges whether or not the contents of registers Ef and Ep have the order registered in execution order table Ep. If these contents do not have the registered order, the operation goes back to Step 32 to wait for the input of the name of the next program. If they have the registered order, Step 34 transfers the data from register En to register Ef, and Step 35 runs the program. After Step 35, the operation returns to Step 32 to wait for the input of the name of the next program.

In the fourth embodiment, since the order of the programs is predetermined, it is impossible to repeat the running of the specific program in an unregistered order to discover the algorithm of the specific program. In this description, the order of only two programs has been predetermined. However, it is possible to predetermine the order of three or more programs.

The present invention can provide a data processing device wherein the running of a specific program which is secret to the user cannot be repeated many times. Thus, the discovery of the algorithm of the specific program is prevented.

Claims:

1. A data processing device comprising:
memory means for memorizing data and programs;
arithmetic operation means for running the memorized programs; and
means for judging whether or not a program to be run by said arithmetic operation means is a specific program, for counting the number of times the specific program is run, and for preventing the specific program from being run over predetermined times.

2. A data processing device comprising:
memory means for memorizing data and programs;
arithmetic operation means for running the memorized programs; and
means for judging whether or not a program to be run by said arithmetic operation means is a specific program and for running a preset delay time program when the specific program is run.

3. A data processing device comprising:
memory means for memorizing data and programs;
arithmetic operation means for running the memorized programs; and
means for judging whether or not a program to be run by said arithmetic operation means is a specific program and for preventing the continuous running of the specific program.

4. The device according to any one of claims 1 to 3, characterized in that said memory means and said arithmetic operation means are constituted by semiconductor integrated circuits.

5. A data processing device comprising:
memory means for memorizing data and programs;
arithmetic operation means for running the memorized programs;
order memorizing means for memorizing the execution order of the programs; and

means for preventing a program from being run by said arithmetic operation means, when the order of running this program does not coincide with the order memorized in said order memorizing means.

6. The device according to claim 5, characterized in that said memory means and said arithmetic operation means are constituted by semiconductor integrated circuits.

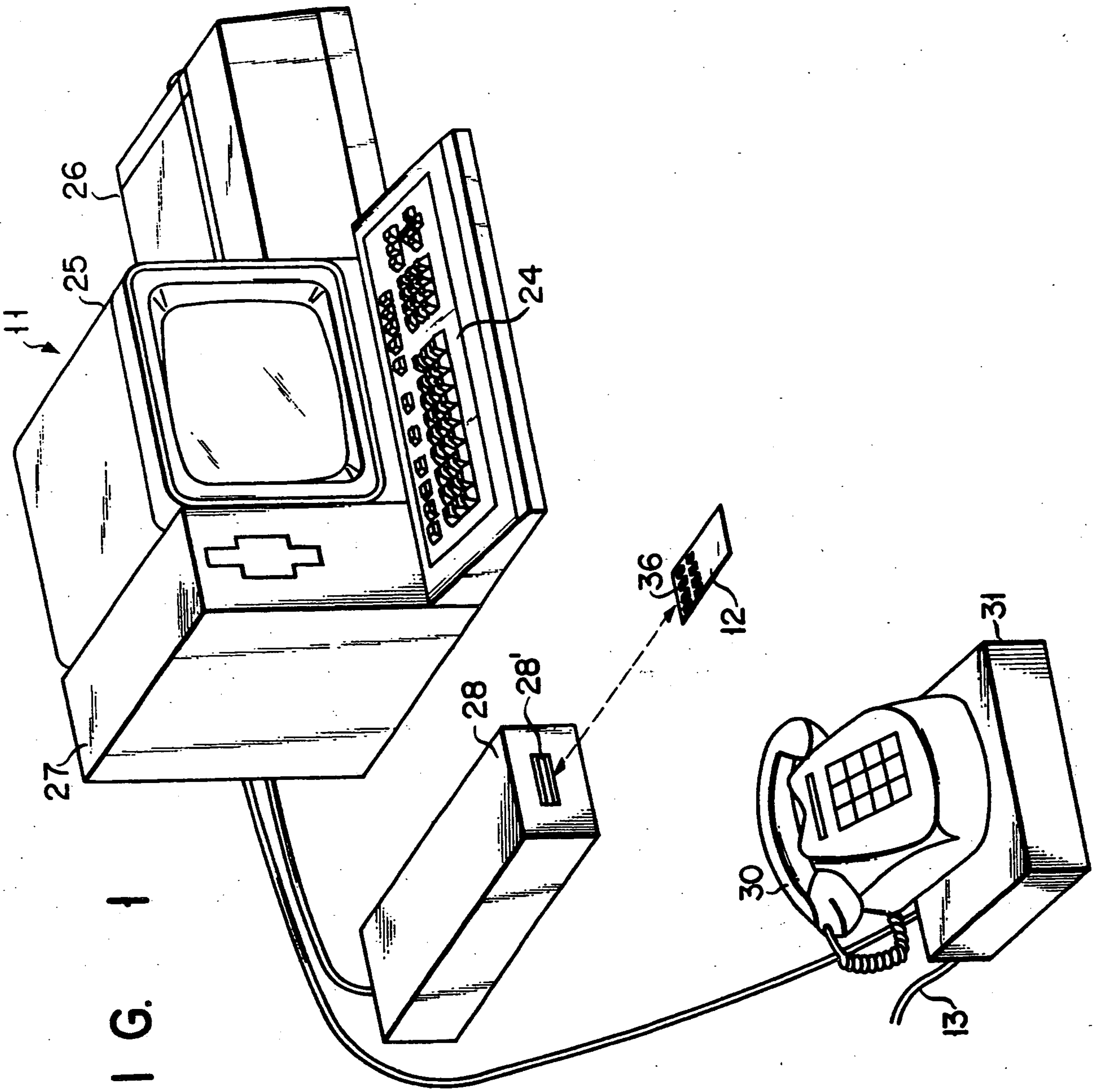
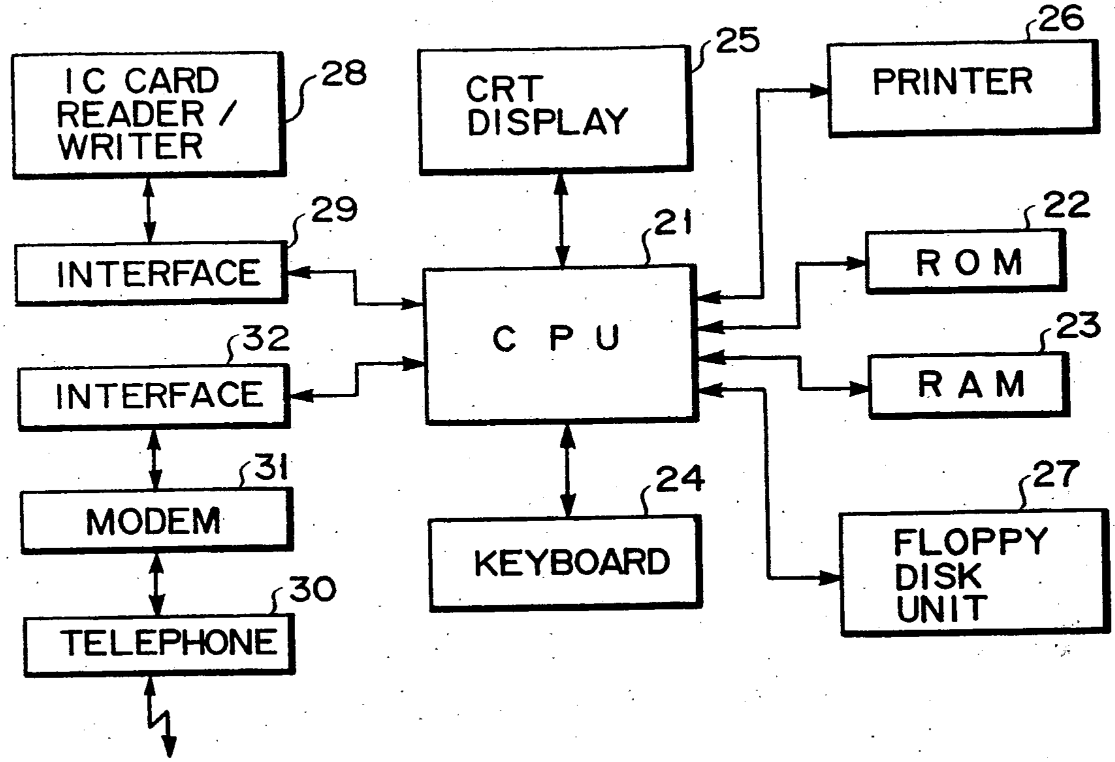


FIG. 1

F I G. 2


IC CARD READER / WRITER
28
CRT DISPLAY
25
PRINTER
26
INTERFACE
29
C P U
21
ROM
22
INTERFACE
32
RAM
23
MODEM
31
KEYBOARD
24
FLOPPY DISK UNIT
27
TELEPHONE
30

F I G. 3

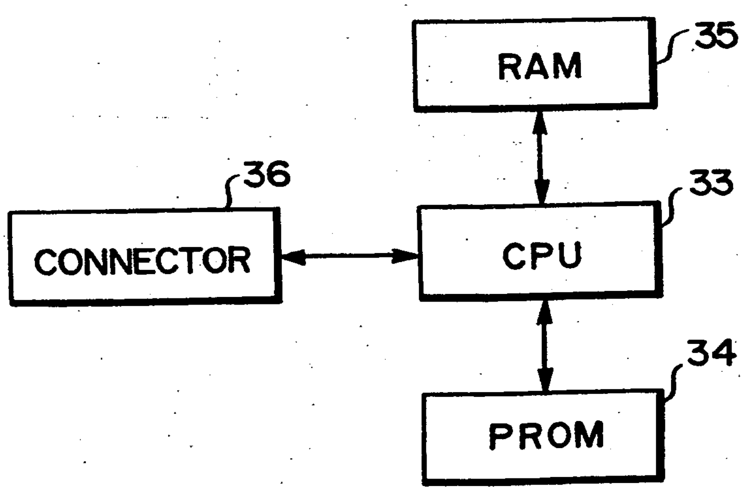

RAM
35
CONNECTOR
36
CPU
33
PROM
34

FIG. 4

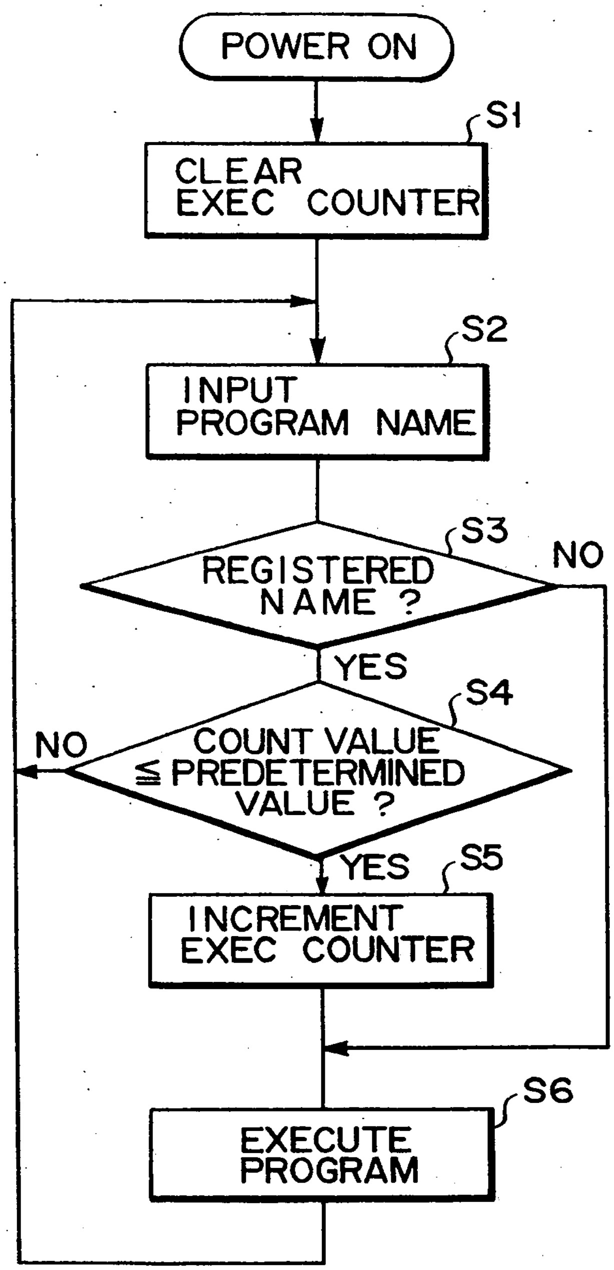
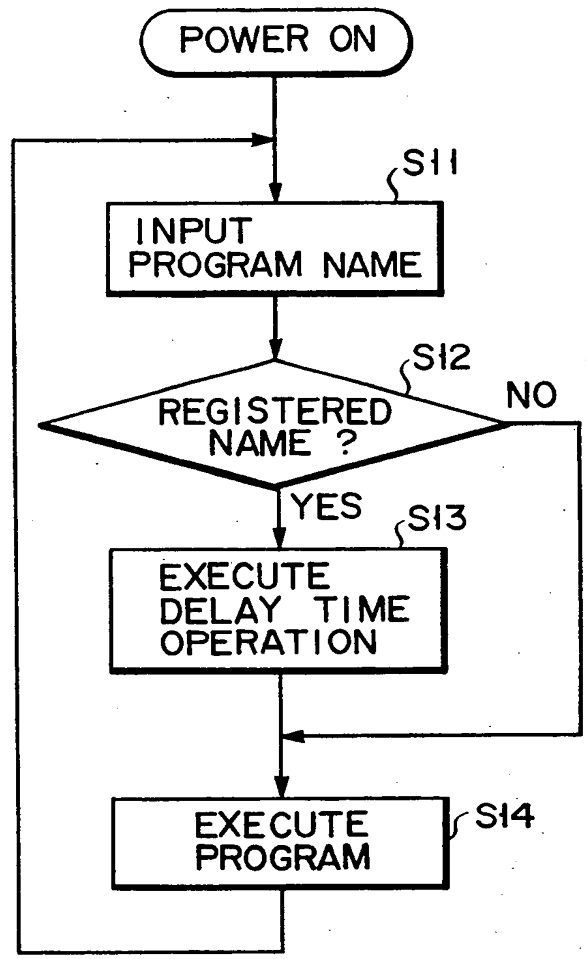
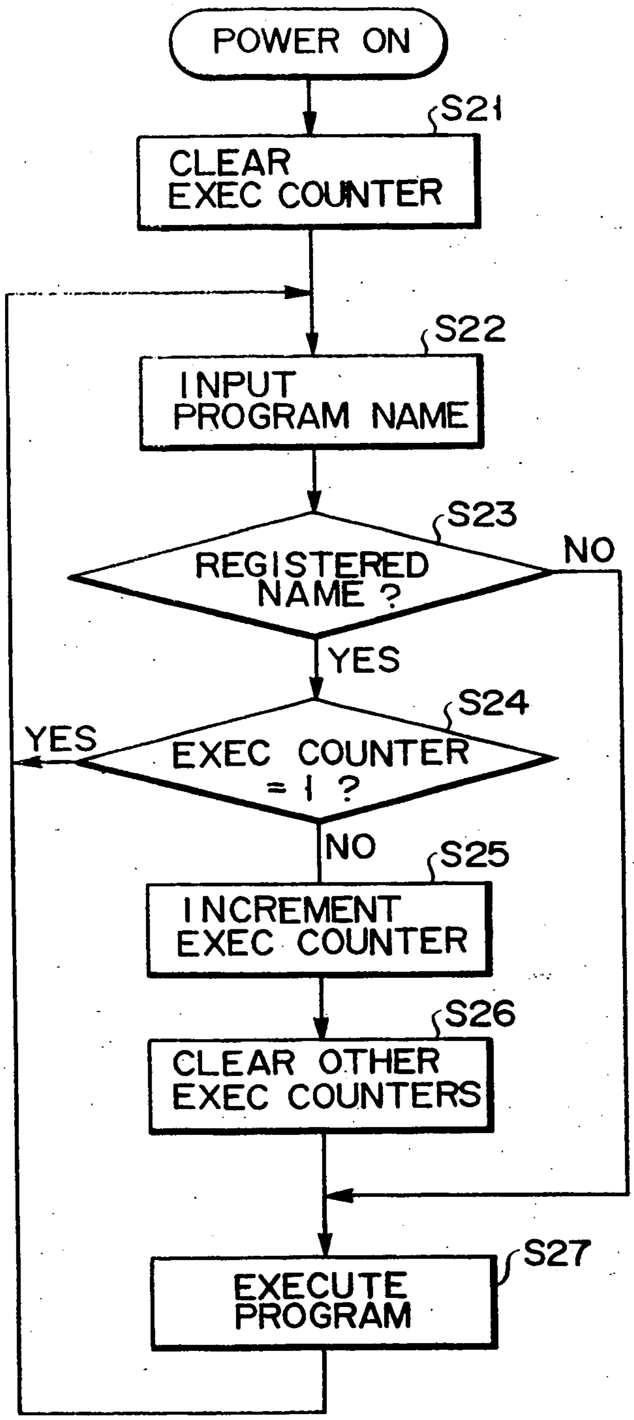

POWER ON
S1
CLEAR EXEC COUNTER
S2
INPUT PROGRAM NAME
S3
REGISTERED NAME ?
NO
YES
S4
COUNT VALUE ≦ PREDETERMINED VALUE ?
NO
YES
S5
INCREMENT EXEC COUNTER
S6
EXECUTE PROGRAM

FIG. 5


POWER ON
S11
INPUT PROGRAM NAME
S12
REGISTERED NAME ?
NO
YES
S13
EXECUTE DELAY TIME OPERATION
S14
EXECUTE PROGRAM

F I G. 6



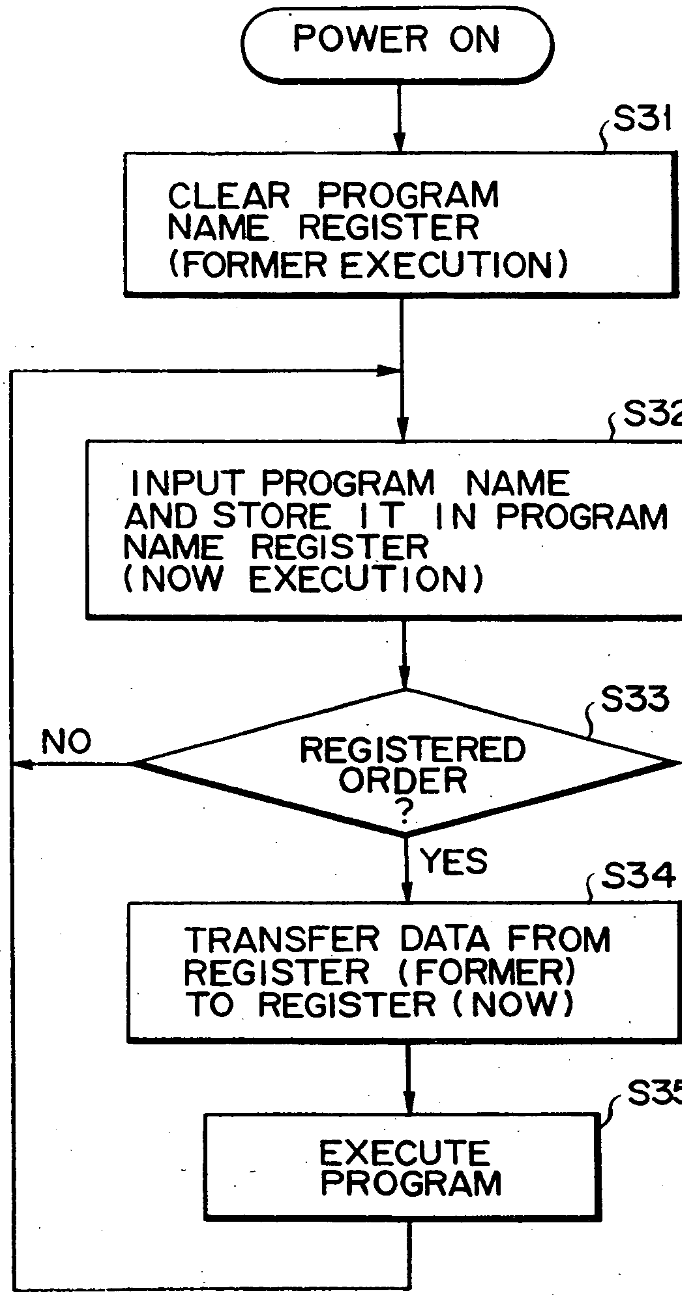
F I G. 7A

E_p	O
	A
	B
	C
	D

F I G. 7B

E_f	O
E_n	A

F I G. 8


POWER ON
S31
CLEAR PROGRAM NAME REGISTER (FORMER EXECUTION)
S32
INPUT PROGRAM NAME AND STORE IT IN PROGRAM NAME REGISTER (NOW EXECUTION)
S33
REGISTERED ORDER ?
NO
YES
S34
TRANSFER DATA FROM REGISTER (FORMER) TO REGISTER (NOW)
S35
EXECUTE PROGRAM

(19) Europäisches Patentamt
European Patent Office
Office européen d s brevets

(11) Publication number: **0 157 303**
A3

(12) # EUROPEAN PATENT APPLICATION

(21) Application number: 85103322.5

(51) Int. Cl.³: G 06 F 1/00

(22) Date of filing: 21.03.85

(30) Priority: 31.03.84 JP 63618/84

(43) Date of publication of application:
09.10.85 Bulletin 85/41

(88) Date of deferred publication of search report: 13.01.88

(84) Designated Contracting States:
DE FR GB

(71) Applicant: Kabushiki Kaisha Toshiba
72, Horikawa-cho Saiwai-ku
Kawasaki-shi Kanagawa-ken 210(JP)

(72) Inventor: Tamada, Masuo c/o Patent Division
K.K. Toshiba 1-1 Shibaura 1-chome
Minato-ku Tokyo 105(JP)

(72) Inventor: Kokuryo, Hitoshi c/o Patent Division
K.K. Toshiba 1-1 Shibaura 1-chome
Minato-ku Tokyo 105(JP)

(72) Inventor: Tamura, Shinsuke c/o Patent Divison
K.K. Toshiba 1-1 Shibaura 1-chome
Minato-ku Tokyo 105(JP)

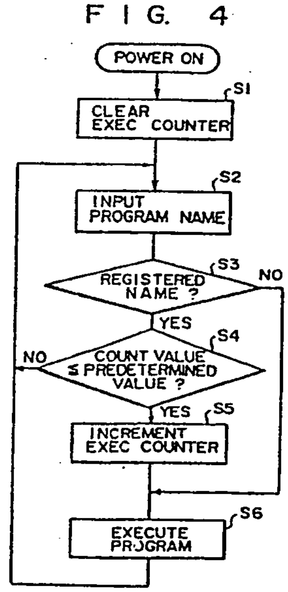
(72) Inventor: Ozaki, Hiroshi c/o Patent Divison
K.K. Toshiba 1-1 Shibaura 1-chome
Minato-ku Tokyo 105(JP)

(74) Representative: Henkel, Feiler, Hänzel & Partner
Möhlstrasse 37
D-8000 München 80(DE)

(54) **Data processing device.**

(57) An IC card comprises CPU (33), PROM (34) for storing programs, and RAM (35) for storing the results of arithmetic operations. PROM (34) stores a table on which the names of specific programs are registered. RAM (35) has execution counters which counts up each time each program is run so that the number of times the specific program is run is memorized. Prior to the running of each program by CPU (33), it is judged whether or not the program is a specific one. If it is the specific program, the execution counter counts up. When the count value of the execution counter is above a predetermined value, the running of the specific program is prevented.

F I G. 4



EP 0 157 303 A3

0157303

European Patent Office

EUROPEAN SEARCH REPORT

Application number

EP 85 10 3322

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.4)
P,Y	PATENT ABSTRACTS OF JAPAN, vol. 8, no. 141 (P-283)[1578], 30th June 1984; & JP-A-59 41 061 (FUJITSU K.K.) 07-03-1984 * Abstract * ---	1	G 06 F 1/00
Y	IEEE SPECTRUM, vol. 21, no. 2, February 1984, pages 43-49, IEEE, New York, US; S.B. WEINSTEIN: "Smart credit cards: the answer to cashless shopping" * Figure 3 *	1	
A	Idem ---	2-6	
A	PATENT ABSTRACTS OF JAPAN, vol. 6, no. 59 (P-110)[937], 16th April 1982; & JP-A-57 726 (USAC DENSHI KOGYO K.K.) 05-01-1982 * Abstract * -----	2-3	TECHNICAL FIELDS SEARCHED (Int. Cl.4) G 06 F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	02-10-1987	MOENS R.A.A.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying th invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited f r other reasons

& : member f the same patent family, corresponding document